

# ***IQMS* PENINGKATAN DERAJAT KONKURENSI EKSEKUSI *QUERY***

**Benhard Sitohang<sup>(1)</sup>**

**Abstract:** Divide Query (*Q*) to primitive as much as we can do, and allocated sub-query(*sQ*) to every *sQ* on exact location where data found, as *Integrated Query Management System (Parallel Execution Control of Sub-Query of Database on Distributed System)* (*IQMS*) based. In this paper will be explained the result of evaluation some *Qs* become *sQ*, and observe the rate of concurrent execution that may be done on *IQMS*, according to system configuration distribution data base used (3 computer: 1 system controller and execution location, and 2 execution locations), by observing data location.

**Keywords:** Distribution Data Base, Relationship Operator, Query, Concurrent Execution, Local Network Computer.

Prinsip dasar strategi alokasi eksekusi *query* (*Q*) pada sistem basis data terdistribusi pada umumnya, adalah: (1) tetap menganggap 1 *Q* sebagai 1 proses (dari sudut pandang sistem operasi), dan (2) eksekusi *Q* pada lokasi data, atau berusaha meminimalkan transfer data antara lokasi (Bitton, 1983; Kom, 1985; Wong, 1976). Prinsip dasar ini secara nyata tidak dapat dicapai, karena kecil kemungkinan bahwa seluruh data yang diperlukan oleh *Q* berada pada lokasi yang sama. Dengan demikian, kemungkinan akan selalu diperlukan pemindahan (transfer) data, dari satu lokasi ke lokasi lain, di mana *Q* dieksekusi, yang berdampak pada waktu eksekusi.

Pada *IQSM*, prinsip dasar no. 2 tersebut di atas tetap dipertahankan untuk *sQ* (eksekusi *sQ* di lokasi data semaksimal mungkin), sedangkan *Q* diupayakan untuk dipecah semaksimal mungkin menjadi *sQ*

primitif (Sitohang, 1992), dengan syarat setiap *sQ* harus dapat dieksekusi sebagai 1 proses (dari sudut pandang sistem operasi), tanpa mengorbankan makna semantik *Q*. Dengan demikian, terdapat 2 kategori persoalan dalam *IQMS*: (1) memecah *Q* menjadi *sQ* semaksimal mungkin tanpa merubah makna semantik *Q*, dan (2) mengalokasikan setiap *sQ* pada lokasi tertentu, di mana data semaksimal mungkin ditemukan. Pada kondisi tertentu, transfer data antara lokasi juga dimungkinkan, agar tidak mengganggu eksekusi *sQ*.

Uji-coba eksekusi konkuren yang dimaksudkan pada tulisan ini, dimaksudkan untuk mengamati sejauh mana *IQMS* dapat melakukan pemecahan *Q* menjadi *sQ* primitif, dan sekaligus mengamati seberapa besar tingkat konkurensi yang dimungkinkan, sesuai dengan karakteristik ketergantungan antara sesama *sQ*.

---

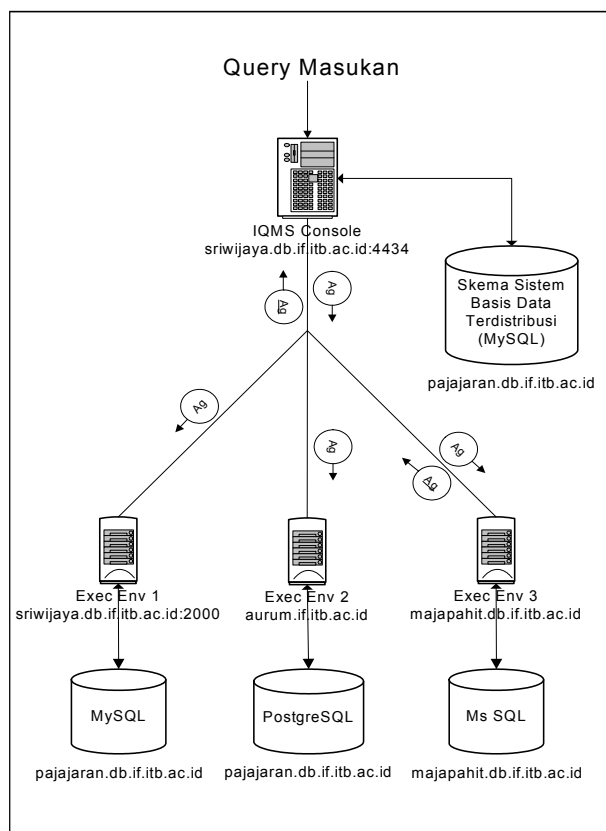
<sup>(1)</sup> Benhard Sitohang, Dr.Ir., Data & Software Engineering Research Division, School of Electrical Engineering & Informatics, Institut Teknologi Bandung, E-mail: benhard@stei.itb.ac.id.

*Prototype IQMS* yang telah dikembangkan (Yudistira, 2002; Sitohang, 2003), dijalankan pada konfigurasi jaringan lokal komputer, telah diuji coba untuk beberapa kasus Q, menggunakan satu skema basis data yang didasarkan pada model ER pada Gambar 2. Hasil uji coba tersebut, menjadi topik bahasan utama pada tulisan ini.

## METODE

### Konfigurasi Uji-coba

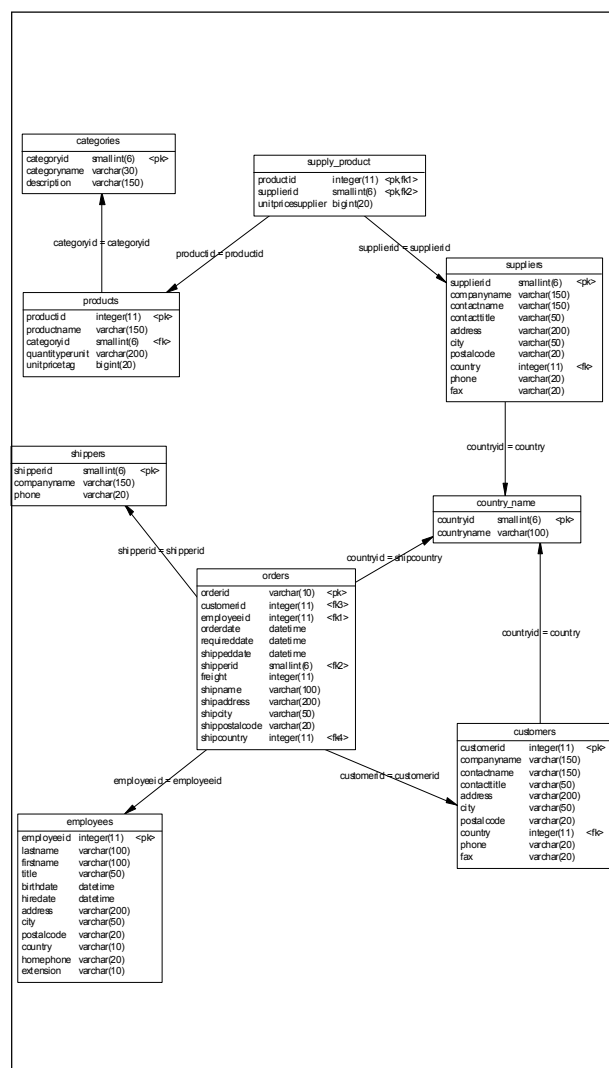
Prinsip eksekusi paralel *sQ* (berbasis operator relasi) pada lingkungan terdistribusi (*distributed data base system*) telah dikembangkan, yang disebut sebagai *IQMS*, dengan arsitektur sebagaimana pada Gambar 1. Terdiri dari 3



Gambar 1 Arsitektur Sistem Komputer *IQMS*

unit komputer masing-masingnya “*single processor*” (*IQMS* : 1 unit sebagai pengendali eksekusi distribusi/*IQMS* dan sebagai peng-eksekusi *sQ*, dan 2 unit sebagai pengeksekusi *sQ* yang dilengkapi dengan *DBMS*, terdiri dari: Sriwijaya, Aurum, dan Majapahit), yang keseluruhannya terhubung dalam 1 jaringan lokal (Yudistira, 2002).

Skema basis data untuk uji coba, adalah sebagaimana disertakan pada Gambar 2, yang terdiri dari 9 relasi (Yudistira, 2002).



Gambar 2 Skema Basis Data

Sedangkan konfigurasi alokasi data dirangkum pada Tabel 1, yang terdiri dari fragmentasi vertikal, horizontal, maupun gabungan keduanya (*hybrid fragmentation*).

Tabel 1 Alokasi Data

Q	Sintaks Q	Keterangan
1	SELECT *	Query tanpa seleksi dan proyeksi data yang tidak terfragmentasi
2	SELECT categoryname FROM categories WHERE categoryid < 5;	Query dengan seleksi dan proyeksi data yang tidak terfragmentasi
3	SELECT *	Query tanpa seleksi dan proyeksi data yang terfragmentasi
4	SELECT employeeid, lastname, firstname FROM employees WHERE country='USA';	Query dengan proyeksi dan seleksi data yang terfragmentasi horizontal
5	SELECT productid, productname, unitpricetag FROM products WHERE unitpricetag > 200000;	Query dengan proyeksi dan seleksi data yang terfragmentasi vertikal
6	SELECT orderid, employeeid, shipcountry FROM orders WHERE shipcountry > 4;	Query dengan proyeksi dan seleksi data yang terfragmentasi hybrid (vertikal-horizontal)
7	SELECT supplierid, companyname, address, city, postalcode FROM suppliers WHERE country = 6;	Query dengan proyeksi dan seleksi data yang terfragmentasi hybrid (horizontal-vertikal)
8	SELECT *	Query dengan <i>cross product</i> antar dua relasi
9.a.	SELECT lastname, firstname, shipcountry FROM employees, orders WHERE employees.employeeid = orders.employeeid;	
9.b.	SELECT companyname, productid, unitpricesupplier FROM suppliers, supply_product WHERE suppliers.supplierid = supply_product.supplierid and unitpricesupplier >= 200000;	Query dengan join
9.c.	SELECT lastname, firstname FROM employees, orders WHERE employees.employeeid = orders.employeeid and customerid < 25 and country = 'UK';	
10	SELECT lastname, homephone FROM employees UNION SELECT contactname, phone FROM customers;	Query dengan union

Pengujian sistem dilakukan terhadap lam bentuk *Q*), sebagaimana tertera pada Tabel 2.

Tabel 2 *Query Untuk Uji-coba*

Q	Sintaks Q	Keterangan
1	SELECT *	Query tanpa seleksi dan proyeksi data yang tidak terfragmentasi
2	SELECT categoryname FROM categories WHERE categoryid < 5;	Query dengan seleksi dan proyeksi data yang tidak terfragmentasi
3	SELECT *	Query tanpa seleksi dan proyeksi data yang terfragmentasi
4	SELECT employeeid, lastname, firstname FROM employees WHERE country='USA';	Query dengan proyeksi dan seleksi data yang terfragmentasi horizontal
5	SELECT productid,productname,unitpricetag FROM products WHERE unitpricetag>200000;	Query dengan proyeksi dan seleksi data yang terfragmentasi vertikal
6	SELECT orderid,employeeid,shipcountry FROM orders WHERE shipcountry > 4;	Query dengan proyeksi dan seleksi data yang terfragmentasi hybrid (vertikal-horizontal)
7	SELECT supplierid,companyname,address, city,postalcode FROM suppliers WHERE country = 6;	Query dengan proyeksi dan seleksi data yang terfragmentasi hybrid (horizontal-vertikal)
8	SELECT * FROM categories, products;	Query dengan <i>cross product</i> antar dua relasi
9.a.	SELECT lastname, firstname,shipcountry FROM employees,orders WHERE employees.employeeid = orders.employeeid;	Query dengan join
9.b.	SELECT companyname, productid, unitpricesupplier FROM suppliers,supply_product WHERE suppliers.supplierid = supply_product.supplierid and unitpricesupplier>=200000;	
9.c	SELECT lastname,firstname FROM employees,orders WHERE employees.employeeid = orders.employeeid and customerid < 25 and country = 'UK';	
10	SELECT lastname, homephone FROM employees UNION SELECT contactname,phone FROM customers;	Query dengan union

### Pengujian Eksekusi *Query*

Pada pengujian, *Q* dimasukkan di lokasi pengendali *IQMS* (Sriwijaya). Di lokasi tersebut, terdapat 2 proses: (1) pemecah *Q* menjadi *sQ* semaksimal mungkin (dilakukan secara terpusat), dengan hipotesa dan algoritma sebagaimana dijelaskan pada (Sitohang, 1992), dan (2) penentuan lokasi eksekusi setiap *sQ* dengan memperhatikan lokasi data pada Tabel 1, tanpa

memperhatikan beban eksekusi setiap komputer (pada konfigurasi uji-coba, alternatif lokasi eksekusi adalah: sriwijaya, Aurum, dan Majapahit).

Dari uji coba eksekusi *Q* dan *sQ*, dilakukan pengamatan untuk 2 hal penting yang menjadi tujuan uji-coba: (1) pemecahan *Q* menjadi *sQ*, dan (2) penentuan lokasi eksekusi *sQ*. Hasil uji-coba dirangkum pada Tabel 3.

**Tabel 3** *Q*, *sQ*, dan Lokasi Eksekusi

Q	QUERY	No.sQ	Sub-Query	Lok. Eksekusi
1	SELECT * FROM categories;	1	SELECT * FROM categories;	Sriwijaya
2	SELECT categoryname FROM categories WHERE categoryid < 5;	1	SELECT categoryname FROM categories WHERE categoryid < 5;	Sriwijaya
3	SELECT * FROM employees;	1	SELECT * FROM employees;	Aurum
		2	SELECT * FROM employees;	Sriwijaya
		3	SELECT * FROM UNION_ALL 1,0;	Sriwijaya
4	SELECT employeeid, lastname, firstname FROM employees WHERE country='USA';	1	SELECT employeeid,lastname,firstname FROM employees WHERE country = 'USA';	Sriwijaya
5	SELECT productid, productname, unitpricetag FROM products WHERE unitpricetag>200000;	1	SELECT productid, productname, unitpricetag FROM products WHERE unitpricetag > 200000;	Majapahit
6	SELECT orderid,employeeid,shipcountry FROM orders WHERE shipcountry > 4;	2	SELECT productid FROM products;	Sriwijaya
		3	SELECT * FROM 1,0 WHERE 1.productid=0.productid;	Sriwijaya
		1	SELECT orderid FROM order_ship WHERE shipcountry > 4;	Majapahit
		2	SELECT orderid,employeeid FROM order_customer;	Majapahit
		3	SELECT orderid,employeeid FROM order_customer;	Aurum
		4	SELECT orderid,employeeid FROM order_customer;	Sriwijaya
		5	SELECT * FROM UNION_ALL 3,2,1;	Sriwijaya
		6	SELECT * FROM 4,0 WHERE 4.orderid=0.orderid;	Sriwijaya
7	SELECT supplierid, companyname, address, city, postalcode FROM suppliers WHERE country = 6;	1	SELECT supplierid,companyname, address, city, postalcode FROM suppliers WHERE country = 6;	Sriwijaya
		2	SELECT supplierid FROM suppliers;	Majapahit
		3	SELECT * FROM 1,0 WHERE 1.supplierid=0.supplierid;	Sriwijaya
8	SELECT * FROM categories, products;	1	SELECT * FROM products;	Majapahit
		2	SELECT * FROM products;	Sriwijaya
		3	SELECT * FROM categories;	Sriwijaya
		4	SELECT * FROM 1,0 WHERE 1.productid=0.productid;	Sriwijaya
		5	SELECT * FROM 3,2;	Sriwijaya
9.a	SELECT lastname, firstname, shipcountry FROM employees, orders WHERE employees.employeeid = orders.employeeid;	1	SELECT shipcountry,orderid FROM order_ship;	Majapahit
		2	SELECT shipcountry,orderid FROM order_ship;	Aurum
		3	SELECT shipcountry,orderid FROM order_ship;	Sriwijaya
		4	SELECT employeeid,orderid FROM order_customer;	Majapahit
		5	SELECT employeeid,orderid FROM order_customer;	Aurum
		6	SELECT employeeid,orderid FROM order_customer;	Sriwijaya
		7	SELECT employeeid, lastname, firstname FROM employees;	Aurum
		8	SELECT employeeid, lastname, firstname FROM employees;	Sriwijaya
		9	SELECT * FROM UNION_ALL 2,1,0;	Sriwijaya
		10	SELECT * FROM UNION_ALL 5,4,3;	Sriwijaya
		11	SELECT * FROM 9,8 WHERE 9.orderid=8.orderid;	Sriwijaya
		12	SELECT * FROM UNION_ALL 7,6;	Sriwijaya
		13	SELECT lastname,firstname,shipcountry FROM 11,10 WHERE 11.employeeid=10.employeeid;	Sriwijaya

Berlanjut...

Lanjutan Tabel 3 Q, sQ, dan Lokasi Eksekusi

Q	QUERY	No.sQ	Sub-Query	Lok. Eksekusi
9.b.	SELECT companyname, productid, unitpricesupplier FROM suppliers, supply_product WHERE suppliers.supplierid = supply_product.supplierid and unitpricesupplier >= 200000;	1	SELECT supplierid FROM suppliers;	Majapahit
		2	SELECT supplierid,companyname FROM suppliers;	Sriwijaya
		3	SELECT supplierid,productid, unitpricesupplier FROM supply_product WHERE unitpricesupplier >= 200000;	Sriwijaya
		4	SELECT supplierid, productid, unitpricesupplier FROM supply_product WHERE unitpricesupplier >= 200000;	Aurum
		5	SELECT supplierid,companyname FROM suppliers;	Aurum
		6	SELECT * FROM 1,0 WHERE 1.supplierid=0.supplierid;	Sriwijaya
		7	SELECT * FROM UNION_ALL 5,4;	Sriwijaya
		8	SELECT * FROM UNION_ALL 3,2;	Sriwijaya
		9	SELECT companyname,productid,unitpricesupplier FROM 7,6 WHERE 7.supplierid=6.supplierid;	Sriwijaya
9.d.	SELECT lastname,firstname FROM employees,orders WHERE employees.employeeid = orders.employeeid and customerid < 25 and country = 'UK';	1	SELECT employeeid,orderid FROM order_customer WHERE customerid < 25;	Sriwijaya
		2	SELECT employeeid,lastname,firstname FROM employees WHERE country = 'UK';	Aurum
		3	SELECT lastname,firstname FROM 1,0 WHERE 1.employeeid=0.employeeid;	Sriwijaya
10	SELECT lastname, homephone FROM employees UNION SELECT contactname, phone FROM customers;	1	SELECT lastname,homephone FROM employees;	Aurum
		2	SELECT lastname,homephone FROM employees;	Sriwijaya
		3	SELECT contactname,phone FROM customers;	Majapahit
		4	SELECT contactname,phone FROM customers;	Aurum
		5	SELECT contactname,phone FROM customers;	Sriwijaya
		6	SELECT * FROM UNION_ALL 1,0;	Sriwijaya
		7	SELECT * FROM UNION_ALL 4,3,2;	Sriwijaya
		8	SELECT * FROM UNION_ALL 6,5;	Sriwijaya

Sumber: Hasil Percobaan

## HASIL DAN PEMBAHASAN

Rekapitulasi pengamatan atas “jumlah sQ” dan “jumlah sQ yang dapat dieksekusi secara paralel/konkuren” berdasarkan uji-coba yang telah dilakukan untuk beberapa Q, dirangkum pada Tabel 4 berikut ini.

Tabel 4 Peningkatan Derajat Paralelisme Eksekusi sQ

Q	Jumlah sQ	Jumlah sQ yang dapat dieksekusi secara paralel
1	1	1
2	1	1
3	3	2
4	1	1
5	3	2
6	6	4
7	3	2
8	5	3
9.a.	13	8
9.b.	9	5
9.c.	3	2
10	8	5

Sumber: Hasil Percobaan

Dari rekapitulasi pengamatan ini (Tabel 4), “jumlah sQ” tidak selalu sama dengan “Jumlah sQ yang dapat dieksekusi secara paralel”. Hal ini disebabkan bahwa tidak semua sQ dapat dieksekusi secara independen dari sQ lainnya. Terdapat kemungkinan, sQ tertentu hanya akan dapat

dieksekusi, berurutan setelah sQ yang lain (ada ketergantungan relatif pada keberadaan data antara sQ dengan sQ lainnya). Prinsip pemecahan Q menjadi sQ, adalah berusaha memilah Q menjadi sQ semaksimal mungkin (sQ primitif), dengan asumsi bahwa sQ primitif hanya terdiri dari 1 operator relasi. Prinsip ini serta algortima pendukungnya, telah dijelaskan secara terperinci pada (Sitohang, 1992; Sitohang, 2002a; Sitohang, 2002b; Sitohang, 2003), termasuk pengertian derajat ketidak-tergantungan sQ relatif pada sQ lainnya sebagai turunan dari Q yang sama.

Sebagai konsekuensinya, Q dapat dipecah menjadi sejumlah sQ, dengan jumlah melebihi jumlah unit komputer yang dialokasikan untuk melakukan eksekusi (dalam kasus uji-coba, jumlah komputer pengeksekusi sQ adalah 3). Sebagai contoh, Query no. 9.a. dapat dipecah menjadi 13 sQ, dan 8 diantaranya dapat dieksekusi secara bersamaan (2 sQ di Majapahit, 3 sQ di Aurum, dan 3 di Sriwijaya),

meskipun komputer yang tersedia untuk itu hanya 3 unit. Dari fakta ini, menambah jumlah komputer yang dapat mengeksekusi  $sQ$  tentunya akan dapat meningkatkan waktu pemrosesan  $Q$  secara keseluruhan. Untuk kasus  $Q$  no. 9.a., tentunya penggunaan sejumlah 8 unit komputer akan dapat memberikan waktu pemrosesan yang terbaik.

Pada uji coba kali ini, pengamatan atas waktu pemrosesan  $Q$  maupun masing-masing  $sQ$  belum dilakukan. Hal ini akan menjadi bagian dari uji-coba tahap selanjutnya. Uji-coba kali ini paling tidak dapat memberikan fakta, bahwa eksekusi paralel beberapa  $sQ$  pada beberapa unit komputer yang terhubung dalam 1 jaringan lokal dapat dilakukan dengan baik, tanpa ada perubahan semantik  $Q$  dan hasil eksekusi sesuai dengan apa yang dimaksudkan pada  $Q$ .

Uji-coba pengamatan atas waktu eksekusi  $sQ$  secara paralel telah dilakukan dan dijelaskan pada (Sitohang, 1992), dan menyajikan reduksi waktu pemrosesan sampai orde 20% (Butar-butur, 1987). Secara teoritis, tentunya hal ini juga berlaku pada uji-coba kali ini (*IQMS*). Kemungkinan distorsi reduksi dapat terjadi, sebagai dampak dari kompleksitas pengendalian jaringan lokal komputer yang digunakan, sebagaimana yang terjadi pada uji-coba sebelumnya (distorsi sebagai dampak dari pengelolaan multi-tasking sistem operasi pada 1 unit komputer). Akan tetapi, adanya peluang eksekusi paralel tersebut akan memberikan reduksi waktu eksekusi yang makin besar, seiring dengan bertambahnya volume data yang ada dalam basis data.

## SIMPULAN

Uji-coba eksekusi paralel *query* pada sistem *IQMS* telah membuktikan 2 hal: (1)  $Q$  dapat dipecah menjadi  $sQ$  sampai pada tingkat primitif (1  $sQ$  hanya

mencakup 1 *operator* relasi), di mana masing-masingnya mempunyai potensi untuk dieksekusi secara paralel, maupun harus berurutan (sebagai dampak dari ketergantungan antara  $sQ$  dengan  $sQ$  lainnya dari  $Q$  yang sama), dan (2) beberapa  $sQ$  yang secara semantik dapat dieksekusi secara paralel, telah dapat dieksekusi secara paralel baik pada 1 unit komputer maupun pada beberapa unit komputer.

Untuk meredusir kompleksitas alokasi data (fragmentasi) pada sistem basis data terpusat, masih dimungkinkan mengalokasikan data hanya di 1 komputer (sebagaimana penggunaan sistem basis data pada 1 unit komputer). Akan tetapi eksekusi  $sQ$  dilakukan oleh beberapa unit komputer yang terhubung dalam jaringan lokal. Degradasi waktu eksekusi mungkin akan ada, sebagai dampak dari pengelolaan jaringan komputer (kecepatan transfer data melalui jaringan lokal). Namun, hal ini akan lebih teratasi di masa mendatang, di mana kecepatan transfer akan makin meningkat.

## RUJUKAN

- Bitton D., Borall H., DeWitt D.J., and Wilkinson W.K. 1983. *Parallel Algorithms for the Execution of Relational Database Operations*, ACM Trans. Database Systems. September 1983.
- Butar-butur M. 1987. *Eksekusi Paralel Operator Relasi SMDR*, Bandung: Jurusan Teknik Informatika – ITB. Februari 1987.
- Kom W., Reiner D.S., Batory D.S. 1985. *Query Processing in Database Systems*. New York: Spring Verlag.
- Sitohang B. 1985. *Eksekusi Operator Relasi RDBMS Pada Sistem Operasi Multi-programming*. KKN – IPKIN. Jakarta. September 1985.
- Sitohang B. 1992. *Eksekusi Paralel Operator Relasi Dari Transaksi Pada Basis Data Model Relasi*. Proceedings ITB. Vol.25, No. 1.
- Sitohang B. 2002. *Parallel Execution of Relational Algebra Operator Under Distributed Database Systems*, ITCC 2002, Las Vegas – Nevada. USA. 8-10 April 2002.
- Sitohang B., and Yudistira D. W. A. 2002. *Eksekusi Paralel Operator Sub-Query Pada Fragment Sistem Basis*

- Data Terdistribusi*. Bandung: Penelitian DPI-FTI-ITB.
- Sitohang B. 2003. *Intergrated Query Management System/ IQMS (Control of Parallel Execution of Sub-Query of database on Distributed System)*, International Conference on Computer, Communication and Control Technologies CCCT03. Orlando – USA. 31 Juli – 1-2 Agustus 2003.
- Wong E., and Youssefi K. 1976. *Decomposition : A Strategy for Query Processing*, ACM Trans. Database Systems. September 1976.
- Yudistira D.W.A. 2002. *Peningkatan Derajat Pemrosesan Paralel Pada Basis Data Terdistribusi Dengan Penentuan Lokasi Eksekusi*, Bandung: Dep. Teknik Informatika - ITB.